

Reasoning-CV: Fine-tuning Powerful Reasoning LLMs for Knowledge-Assisted Claim Verification

Zhi Zheng

National University of Singapore
zhengzhi@comp.nus.edu.sg

Wee Sun Lee

National University of Singapore
leews@comp.nus.edu.sg

Abstract

Claim verification is essential in combating misinformation, and large language models (LLMs) have recently emerged in this area as powerful tools for assessing the veracity of claims using external knowledge. Existing LLM-based methods for claim verification typically adopt a *Decompose-Then-Verify* paradigm, which involves decomposing complex claims into several independent sub-claims and verifying each sub-claim separately. However, this paradigm often introduces errors during the claim decomposition process. To mitigate these errors, we propose to develop the *Chain-of-Thought (CoT)-Verify* paradigm, which leverages LLM reasoning methods to generate CoT-verification paths for the original complex claim without requiring decompositions into sub-claims and separate verification stages. The *CoT-Verify* paradigm allows us to propose a natural fine-tuning method called Reasoning-CV to enhance the verification capabilities in LLMs. Reasoning-CV includes a supervised fine-tuning (SFT) stage and a self-improvement direct preference optimization (DPO) stage. Utilizing only an 8B pre-trained LLM, Reasoning-CV demonstrates superior knowledge-assisted claim verification performances compared to existing *Decompose-Then-Verify* methods, as well as powerful black-box LLMs such as *GPT-4o+CoT* and o1-preview. Our code is available¹.

1 Introduction

Claim verification (Aly et al., 2021; Jiang et al., 2020) (also known as fact-checking (Eldifrawi et al., 2024)) is a crucial task that involves evaluating the veracity of a complex claim using provided or retrieved knowledge as evidence. This task plays a vital role in curbing the spread of misinformation (Tambuscio et al., 2015; Min et al.,

2023) and is instrumental in identifying hallucinations (Huang et al., 2023; Nie et al., 2024) of generative models. With the recent advancements in Large Language Models (LLMs) (Touvron et al., 2023; Achiam et al., 2023; Liu et al., 2024), LLM-based approaches (Wang and Shu, 2023; Zhao et al., 2024) have demonstrated exceptional performance in the knowledge-assisted claim verification task.

Claim: Cristiano Ronaldo and his team lost in all their international matches.

 ↓ Decompose

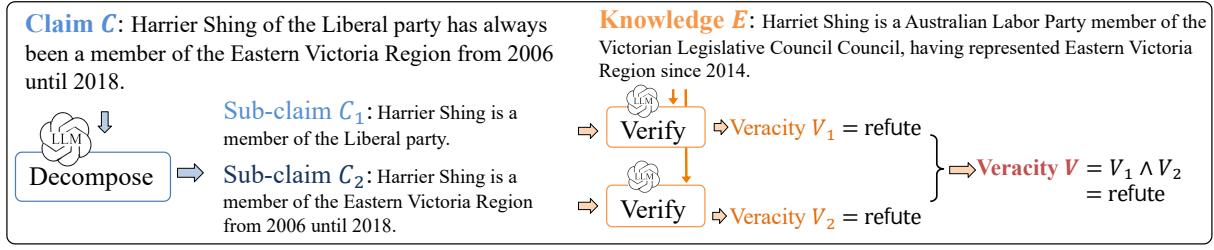
Sub-claim 1: Cristiano Ronaldo has a team. (**Error!**
Over-decomposition, changed the original meaning)

Sub-claim 2: Cristiano Ronaldo and his team lost all their international matches.

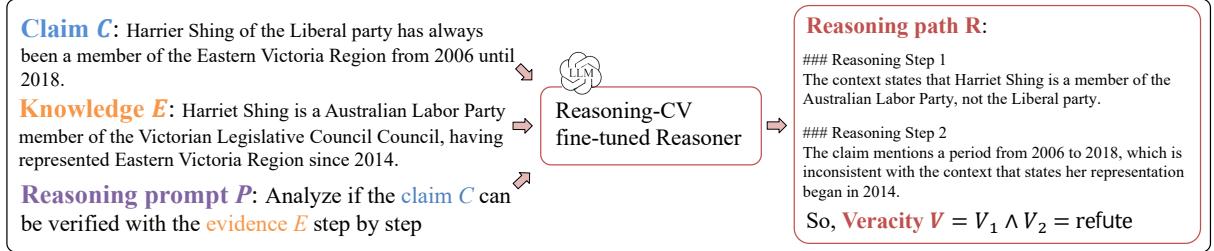
Figure 1: An example of claim decomposition using *GPT-4o* and prompts from Min et al. (2023). In this case, we get a redundant sub-claim (i.e., Sub-Claim 1 in Figure) that changes the original meaning.

As an intricate task, claim verification requires a precise understanding of complex claims and their associated knowledge. So, existing LLM-based methods generally adopt a *Decompose-Then-Verify* paradigm (Hu et al., 2024) to reduce the complexity of claim verification. As shown in Figure 2(a), these methods begin by leveraging the text-processing capabilities of LLMs to decompose a complex claim into several independent sub-claims. Then, LLMs are employed again to judge the veracity of each simple sub-claim, taking provided or retrieved knowledge as evidence (Wang and Shu, 2023). Finally, *Decompose-Then-Verify* methods will judge the claim as correct if all its sub-claims are supported. Although this paradigm can simplify the verification of complex claim sentences, LLM-based decomposition will inevitably lead to errors, obtaining uncertain or wrong sub-claims (Hu et al., 2024) (e.g., the case in Figure 1), which

¹The code of Reasoning-CV is available at <https://github.com/zz1358m/Reasoning-CV>.



(a) *Decompose-Then-Verify* paradigm for knowledge-assisted claim verification



(b) *CoT-Verify* for end-to-end knowledge-assisted claim verification (Ours)

Figure 2: Existing LLM-based claim verification methods generally adopt a *Decompose-Then-Verify* paradigm (a). These methods first break a complex claim into several independent sub-claims, then leverage provided or retrieved knowledge to judge the veracity of each simple sub-claim. We propose to use a *CoT-Verify* paradigm (b), which aims at directly generating high-quality CoT-verification paths for the veracity of complex claims. It can reduce the number of LLM calls, eliminate the decomposition error, and achieve significantly better accuracy after the proposed Reasoning-CV fine-tuning.

will significantly impact the verification accuracy.

In this paper, we observe that when employing the Chain-of-Thought (CoT) prompt strategy (Wei et al., 2022; Sprague et al., 2024) for claim verification, advanced LLMs such as *GPT-4o* will naturally perform verification analysis on each of the key facts among the claim in its CoT reasoning path (See Appendix B for examples). This observation indicates that LLMs have the potential to effectively analyze complex claims with a *CoT-Verify* process in a single LLM call, and explicitly decomposing claims into separate verification stages is unnecessary. Compared to *Decompose-Then-Verify*, the *CoT-Verify* paradigm can reduce the number of LLM calls while mitigating the possible errors during the claim decomposition process.

To develop powerful reasoning LLMs for *CoT-Verify*, this paper proposes a novel **Reasoning-CV** (Reasoning-Claim-Verification) fine-tuning method for open-source LLMs. As shown in Figure 3, the proposed Reasoning-CV method is a **two-stage fine-tuning process**. In the first stage, we prompt *GPT-4o* to generate reliable reasoning paths for ground-truth labels and do supervised fine-tuning (SFT) on pre-trained LLMs. In the second stage, we design a self-improvement direct preference optimization (DPO) (Rafailov et al., 2023) pro-

cess, gradually guiding the fine-tuned LLM to update the consistency and judgment of its reasoning path. We evaluate the proposed Reasoning-CV on a wide collection of knowledge-assisted claim verification test sets and benchmarks, and LLMs fine-tuned with Reasoning-CV demonstrate superior performance compared to existing claim verification methods and frontier LLMs including *GPT-4o* and o1-preview, utilizing only 8B or fewer parameters. Our contributions are summarized as follows:

- This paper first proposes to build the claim verification process in a long CoT reasoning path and presents Reasoning-CV to fine-tune LLMs for high-quality reasoning paths.
- The proposed Reasoning-CV presents a novel method for generating high-quality DPO training data, which is used to fine-tune LLMs in an iterative self-improvement manner. Instead of simply generating CoT-verification paths and checking whether they agree with the ground truth verification label, we provide the possible labels and ask the LLM to generate a CoT that agrees with each label. Conditioning on the label to generate allows LLMs to generate higher-quality CoTs for correct labels and confusing CoTs (that we want

to learn not to generate) for incorrect labels.

- Reasoning-CV can achieve superior knowledge-assisted claim verification performance compared to existing methods and frontier LLMs, using only a *Meta-LlaMA-3-8B-Instruct* base LLM.

2 Related Work

2.1 Task Definition: Claim Verification

The claim verification task aims to determine the veracity V of a claim C based on knowledge E . There are two settings for the source of knowledge E , gold evidence, and open book (Aly et al., 2021). In the **gold evidence** setting, each claim is provided with knowledge that can determine its veracity, while the **open book** setting requires verification methods to retrieve knowledge from sources based on the claim C . There are also two settings for the range of veracity V , that is, **w/o NEI** and **w NEI**. Under the **w/o NEI** setting, the veracity is predicted from ‘support’ and ‘refute’, while the **w NEI** setting introduces another ‘not enough evidence’ (abbreviated as ‘NEI’) option. Some datasets, like FEVEROUS (Aly et al., 2021), have no ‘NEI’ label and support only the **w/o NEI** setting (Jafari and Allan, 2024), while other datasets (i.e., Healthver (Mourad Sarrouti and Demner-Fushman, 2021)) can support the **w NEI** setting.

2.2 Decompose-Then-Verify Paradigm

The *Decompose-Then-Verify* paradigm is widely adopted in LLM-based claim verification methods (Hu et al., 2024), including FACTSCORE (Min et al., 2023), SAFE (Wei et al., 2024), PACAR (Zhao et al., 2024), etc. (Wang and Shu, 2023; Zhang and Gao, 2023; Jafari and Allan, 2024). Their verification processes involve breaking down the claim C into a set of sub-claims $\mathcal{S} = \{c_1, c_2, \dots, c_p\}$ with LLMs and using a verifier (usually an LLM-based verifier (Zhao et al., 2024)) to assess the veracity v_i of each sub-claim c_i , $i \in \{1, \dots, p\}$. Finally, individual verification results are aggregated to produce a final judgment of veracity V . For example, in FOLK (Wang and Shu, 2023), the final veracity is processed as the conjunction paradigm for the veracity of sub-claims as follows:

$$V \leftarrow v_1 \wedge v_2 \wedge \dots \wedge v_p. \quad (1)$$

Compared to directly verifying the claim with reasoning steps (i.e., *CoT-Verify*), *Decompose-Then-*

Verify methods can avoid verifying complex claim sentences (Min et al., 2023). However, as shown in recent research (Hu et al., 2024; Tang et al., 2024a), these methods introduce decomposition errors, which can result in generating uncertain or inconsistent sub-claims (as shown in Figure 1). Once making an incorrect decomposition, even the optimal sub-claim verification results may achieve an incorrect veracity judgment V , which will significantly impact the claim verification performance.

2.3 Other Paradigms

Besides the *Decompose-Then-Verify* paradigm, there are also claim verification methods with an adaptive retrieval and verification framework (Pan et al., 2023; Shao et al., 2023; Quelle and Bovet, 2024). Similar to adaptive retrieval methods for Retrieval-augmented generation (RAG) (Gao et al., 2023; Asai et al., 2023), these methods first use the original claim C to retrieve evidence for verification and then repeatedly handle the uncertain parts of the claim in the previous verification step by retrieving new knowledge as verification evidence. Compared to these methods, Reasoning-CV focuses on fine-tuning LLMs to make the first verification process sufficiently reliable.

Directly employing the pre-trained LLMs with CoT prompts (e.g., *GPT-4o+CoT*) or using black-box reasoning LLMs (e.g., o1-preview) are existing works under the *CoT-Verify* paradigm. Some pre-trained reasoning models like o1-preview are proficient in handling complex claims C (Sprague et al., 2024) without requiring decomposition or iterative processes, albeit at a higher cost.

2.4 Fine-Tuning LLMs for Claim Verification

Several claim verification methods propose fine-tuning techniques on neural models for counter-example generation (Zhu et al., 2023), evidence retrieval (Zhang and Gao, 2024; Huang et al., 2024), or veracity generation between sub-claims and evidence (Zeng and Zubiaga, 2024; Tang et al., 2024a). Among these, Minicheck (Tang et al., 2024a) is closely related to our work, as it fine-tunes LLMs mainly for the verification step in the *Decompose-Then-Verify* framework, focusing on verifying simple claims based on provided evidence. In contrast, our proposed Reasoning-CV fine-tuning method enhances the reasoning abilities of LLMs, enabling them to tackle **complex** claims directly without decomposition, thereby avoiding errors explicitly associated with the decomposition process.

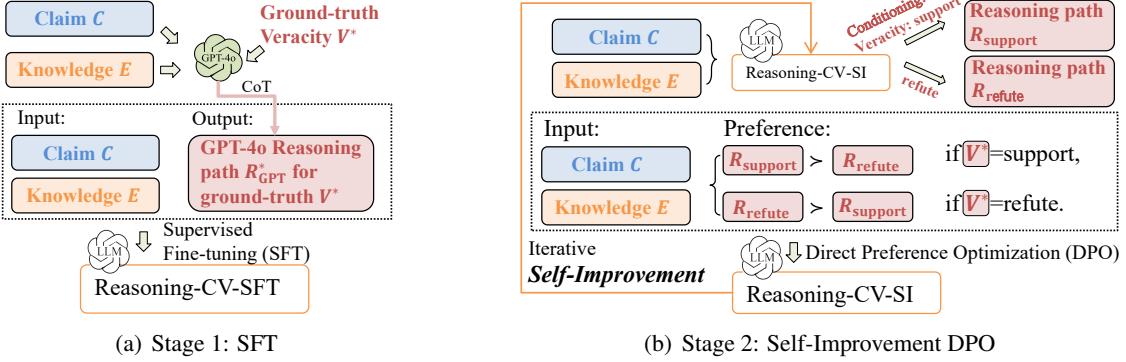


Figure 3: The proposed Reasoning-CV is a two-stage fine-tuning method for knowledge-assisted claim verification. Given a training dataset with Claim C , Knowledge E , and ground-truth veracity V^* , Reasoning-CV can obtain LLMs with high-quality CoT-verification paths. The first stage aims at distilling the $GPT\text{-}4o$ generated reasoning path for ground-truth veracity V^* , and the second stage is designed to iteratively enhance the judgment and consistency of reasoning paths from the fine-tuned LLMs.

Dataset	Domain	Claim	Evidence	Veracity Label $V^* \in$	Training data number
FEVEROUS	Wikipedia	Brief, Complex	Tabular, Textual	{support,refute}	26,828
		2-hop claims			9,006
HOVER	Wikipedia	3-hop claims	Textual	{support, refute}	6,053
		4-hop claims (complex)			3,012
Healthver	COVID-19	Brief, Complex	Textual	{support, NEI, refute}	10,590
Total	All domains	Mixed Hardness	Language	w NEI and w/o NEI	55,489

Table 1: The collection of the training dataset for Reasoning-CV. The Reasoning-CV training dataset covers three datasets with complex claims. Specifically, the Healthver dataset has a ‘NEI’ option.

3 Reasoning-CV: Obtaining Advanced Reasoning LLMs for *CoT-Verify*

This paper proposes a two-stage fine-tuning method, Reasoning-CV, to develop powerful reasoning LLMs for knowledge-assisted claim verification. To optimize the quality of *CoT-Verify* reasoning paths for both *w NEI* and *w/o NEI* settings, we first build a training dataset containing **55K** pairs of **claims C** with different hardness and their **gold evidence knowledge E** in text or tabular form. As shown in Table 1, the training data is collected from three well-known claim verification datasets with complex claims (that is, HOVER (Jiang et al., 2020), FEVEROUS (Aly et al., 2021), and Healthver (Mourad Sarrouti and Demner-Fushman, 2021)). Among them, we include HOVER and FEVEROUS for the *w/o NEI* setting and Healthver for the *w NEI* setting.

As shown in Figure 3, the proposed Reasoning-CV is a two-stage fine-tuning process. In the first stage, it performs SFT on an open-source LLM to distill the CoT-verification path from $GPT\text{-}4o$. We provide $GPT\text{-}4o$ the ground-truth veracity (noted as

V^*) for reliable CoT-verification paths. In the second stage, we ask the fine-tuned LLM to generate reasoning paths for each veracity option and use the generated reasoning paths to build preference pairs for DPO fine-tuning, emphasizing reasoning paths with the ground-truth veracity.

3.1 First Stage: Reasoning Path SFT

Compared to advanced black-box LLMs like $GPT\text{-}4o$, pre-trained open-source LLMs like the 8B LlaMA LLM tend to generate inferior reasoning paths (Refer to the *Meta-LlaMA-3-8B-Instruct + CoTPrompt* results in Table 2). So, the first stage of Reasoning-CV fine-tunes open-source LLMs with reliable reasoning paths generated by $GPT\text{-}4o$ to build their *CoT-Verify* capability.

To generate reliable $GPT\text{-}4o$ reasoning paths for claim C and knowledge E , we require $GPT\text{-}4o$ to generate a CoT that agrees with the ground-truth label V^* . The generated reasoning path can be regarded as $GPT\text{-}4o$ ’s CoT-explanation of the ground-truth label V^* . In our implementation, we prompt $GPT\text{-}4o$ with P_{CE} for claim C and knowledge E , and a conditioning prompt P_{V^*} providing

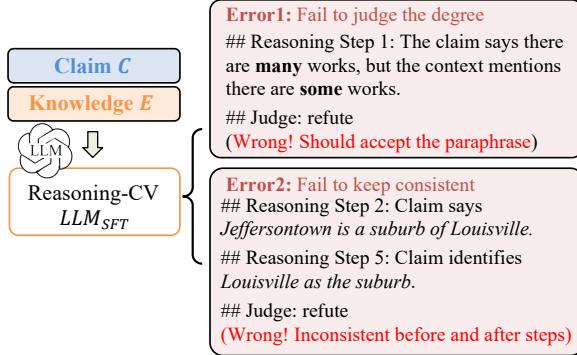


Figure 4: A qualitative study for the possible error of reasoning LLMs after the first stage of Reasoning-CV fine-tuning. LLMs still face difficulties in judging the acceptance threshold (e.g., **Error1**) and in maintaining consistency for their long reasoning path (e.g., **Error2**).

the ground truth label V^* . The *GPT-4o* reasoning path R_{GPT}^* is generated as follows (Refer to Appendix A for detailed prompts and instructions):

$$R_{GPT}^* = \text{GPT-4o}(P_{CE}, P_{V^*}). \quad (2)$$

We enumerate data in the training dataset (i.e., Table 1), forming a dataset of reasoning paths R_{GPT}^* . Although provided with the ground-truth veracity V^* , *GPT-4o* will still generate reasoning paths with incorrect judgments. So, we conduct a data cleaning process that removes all these reasoning paths that do not match the label (i.e., $V^* \notin R_{GPT}^*$) and ultimately obtain 50,011 reliable paths. We use SFT to fine-tune LLMs over reliable *GPT-4o* reasoning paths for three epoches and note the LLM after the first stage of Reasoning-CV as LLM_{SFT} .

3.2 Second Stage: Self-Improvement DPO

As shown in Figure 4, after the SFT stage, the fine-tuned LLM_{SFT} will still have difficulties in: **1) Judging the acceptance threshold.** For example, in Case 1 of Appendix C, the knowledge contains a fact using the adverb "some" to show its degree, and the claim paraphrases the word "some" to "many". Human judgment may accept this paraphrase, but LLM_{SFT} incorrectly judges it as 'refute'. **2) Keeping consistent in the former and latter parts of the reasoning path.** When processing complex claims, for example, in Case 2 of Appendix C, the reasoning path may initially provide correct assertions (Step 2). However, in step 5, LLM_{SFT} incorrectly changes its previous assertions, resulting in inconsistency and a wrong judgment.

We believe both drawbacks stem from the fact that the generated *GPT-4o* reasoning paths R_{GPT}^*

are not optimal for the *CoT-Verify* of claims. So, to further improve the judgment and consistency of fine-tuned LLMs, we propose the second stage of Reasoning-CV, iteratively building preferences for correct over wrong reasoning paths.

In the second stage of Reasoning-CV, we provide the fine-tuned LLM with all possible veracity options (i.e., 'support' and 'refute' for data in FEVEROUS and HOVER; 'support', 'NEI', and 'refute' for data in Healthver) and ask the LLM to conditionally generate reasoning paths for each option. Then, we utilize the generated paths to fine-tune LLMs with the DPO technique (Rafailov et al., 2023), preferring the reasoning path with the ground-truth veracity over incorrect ones. By employing LLMs after the DPO fine-tuning to generate new preference pairs, this conditional-generation-based DPO process can be conducted in a self-improving manner. So, starting from $LLM_{SI_0} = LLM_{SFT}$, in the round $i \in \{1, 2, \dots\}$, we prompt the fine-tuned LLM $LLM_{SI_{i-1}}$ with P_{CE} and the conditioning prompt (e.g., $P_{support}$, P_{refute}) to generate the preference dataset $\mathcal{D}_{SI_{i-1}}$ for DPO as follows (taking claims for FEVEROUS and HOVER as examples, which have only two veracity options, 'support' and 'refute'):

$$\begin{aligned} R_{support} &= LLM_{SI_{i-1}}(P_{CE}, P_{support}), \\ R_{refute} &= LLM_{SI_{i-1}}(P_{CE}, P_{refute}). \end{aligned} \quad (3)$$

$$\begin{aligned} \mathcal{D}_{SI_{i-1}} = \\ \left\{ \begin{array}{l} [V^* \in R_{support}] (R_{refute} \prec R_{support}), \\ [V^* \in R_{refute}] (R_{support} \prec R_{refute}) \end{array} \right\}. \end{aligned} \quad (4)$$

Similar to the first stage, we use the 55K Reasoning-CV training dataset for claims, evidence, and veracity labels in each round. For the Healthver dataset, we enumerate the 'support', 'NEI', and 'refute' options and Appendix D.1 shows the details of constructing the preference $\mathcal{D}_{SI_{i-1}}$ for Healthver. The fine-tuned LLMs' explanation of incorrect veracities tends to be forced and inconsistent, and using DPO to discourage such reasoning paths can help LLMs learn the thresholds for judgments and help improve the consistency of the *CoT-Verify* generation. We conduct two rounds of self-improvement by default ($i \in \{1, 2\}$), and in each round of obtaining LLM_{SI_i} , we fine-tune $LLM_{SI_{i-1}}$ on the preference dataset of paths $\mathcal{D}_{SI_{i-1}}$ for three epoches.

Gold Evidence								
Dataset	Param Size	FEVEROUS	HOVER (2-hop)	HOVER (3-hop)	HOVER (4-hop)	Healthver		Avg.
Method	Settings	w/o NEI	w/o NEI	w/o NEI	w/o NEI	w/o NEI	w NEI	
Meta-LLaMA-3-70B-Instruct	70B	91.17	76.28	73.15	69.01	71.44	<u>64.44</u>	74.25
GPT-4o	N/A	91.64	79.48	75.83	73.52	67.87	58.37	74.45
GPT-4o + CoT Prompt	N/A	90.80	81.44	76.32	74.86	70.26	52.64	74.39
o1-preview	N/A	89.43	<u>83.27</u>	<u>78.69</u>	<u>79.59</u>	68.84	62.32	<u>77.02</u>
Minicheck-7B	7B	89.54	82.62	71.68	67.04	55.63	-	73.30
Established Decompose-Then-Verify Methods								
PACAR	N/A	<u>94.43*</u>	76.86*	70.10*	69.95*	-	-	-
FactScore	N/A	85.55	79.00	68.12	62.49	56.31	60.93	68.73
Decompose + Minicheck-1B	1B	84.80	75.47	61.67	58.81	47.35	-	65.62
Decompose + Minicheck-7B	7B	85.66	78.39	61.67	53.11	49.53	-	65.67
Reasoning-CV for High-Quality CoT-Verify								
Meta-LLaMA-3-8B-Instruct	8B	90.34	74.24	67.21	66.39	<u>71.72</u>	55.48	70.90
+ CoT Prompt	8B	87.06	74.10	67.29	64.09	62.19	52.06	67.80
+ Reasoning-CV-LLM _{SFT}	8B	93.72	81.26	78.70	75.65	73.10	64.19	77.77
+ Reasoning-CV-LLM _{SI₁}	8B	95.22	84.01	83.06	81.01	76.40	67.72	81.24
+ Reasoning-CV-LLM _{SI₂}	8B	95.50	85.97	83.93	83.02	78.41	65.59	82.07

Table 2: Results of Reasoning-CV fine-tuned LLMs in in-domain test sets with gold evidence considering both *w NEI* and *w/o NEI* settings. Avg. shows the average performance of methods on different test sets and settings. We **bold** the best-performing LLM on each test set and underline the best-performing baseline. Results with * are collected from Zhao et al. (2024).

4 Experiment

In this section, we apply the proposed Reasoning-CV fine-tuning method to open-source LLM *Meta-LlaMA3.1-8B-Instruct* and evaluate the ability of Reasoning-CV to obtain reasoning LLMs for powerful knowledge-assisted *CoT-Verify*.

Dataset In this section, we include three in-domain test sets from datasets with complex claims discussed in Table 1, i.e., **FEVEROUS** (Aly et al., 2021), **HOVER** (Jiang et al., 2020), and **Healthver** (Mourad Sarrouti and Demner-Fushman, 2021). Meanwhile, to evaluate the generalization ability of the fine-tuned LLMs on claim-evidence pairs of different domains, we also include **LLM-AggrFact** (Tang et al., 2024a), **SciFact** (Wadden et al., 2020), and **VitaminC** (Schuster et al., 2021) datasets for out-of-domain. These datasets consist of relatively simple claims collected from a wide range of fields, including LLM generations (Tang et al., 2024a) and science (Wadden et al., 2020), which are rarely covered in the Reasoning-CV training dataset. Appendix F.1 includes details for these test sets.

Implementation Details As mentioned in Section 2.1, FEVEROUS, HOVER, and LLM-AggrFact only support the *w/o NEI* setting. Datasets Healthver, VitaminC, and SciFact have the ‘NEI’ option in their label, so they can adopt both the *w NEI* setting and the *w/o NEI* setting (by considering the ‘NEI’ label as a special case of ‘refute’). For claim verification under the two

settings, we utilize the same reasoning LLM with different prompts shown in Appendix A. We implement both stages of Reasoning-CV fine-tuning on an H100-96GB GPU with LoRA (Hu et al., 2022). The LoRA rank is 64 in our experiment, and the learning rates for the two stages are $5e^{-5}$ and $5e^{-6}$, respectively.

Baseline and Metrics This section involves a wide range of knowledge-assisted claim verification methods as the baselines, which include 1) Advanced LLMs and black-box reasoning models, i.e., **Minicheck** (Tang et al., 2024a), **GPT-4o**, **GPT-4o+CoT**, and **o1-preview**. 2) Claim verification methods with *Decompose-Then-Verify* paradigm, e.g., **FactScore** (Min et al., 2023), **PACAR** (Zhao et al., 2024), and **Decompose + Minicheck** (Tang et al., 2024a). For Decompose + Minicheck, we first decompose complex claims using prompts in Min et al. (2023), then introduce Minicheck (Tang et al., 2024a) fine-tuned LLMs to verify each of them. Please refer to Appendix F.2 for the implementation details of baselines.

Following Zhao et al. (2024), this paper uses the Macro-F1 score as the evaluation metric to assess the methods’ performance on test sets. To better exhibit the performance difference between methods, we multiply each Macro-F1 score by 100.

4.1 In-Domain Performance

We first investigate the effectiveness of the proposed Reasoning-CV on in-domain test sets

		Gold Evidence									
Dataset	Param Size	LLM-AggrFact	SciFact-train		SciFact-dev		VitaminC-dev		VitaminC-test		Avg.
Method	Settings	w/o NEI	w/o NEI	w NEI	w/o NEI	w NEI	w/o NEI	w NEI	w/o NEI	w NEI	
GPT-4o	N/A	79.40	88.99	77.57	85.57	<u>75.50</u>	81.48	65.21	83.37	67.02	78.23
o1-preview	N/A	-	86.53	81.44	86.44	79.72	-	-	-	-	-
Minicheck-1B	1B	67.86	81.87	-	82.57	-	71.66	-	73.77	-	-
Minicheck-7B	7B	76.58	86.71	-	<u>87.21</u>	-	78.52	-	78.87	-	-
Reasoning-CV for High-Quality <i>CoT-Verify</i>											
Meta-LLaMA-3-8B-Instruct	8B	74.58	86.34	68.89	83.14	67.08	75.50	53.37	77.40	55.37	71.30
+ CoT Prompt	8B	73.61	81.31	66.56	83.81	62.19	78.05	62.10	78.93	63.31	72.21
+ Reasoning-CV- <i>LLM_{SFT}</i>	8B	76.21	86.19	78.99	87.46	80.75	82.38	68.22	83.91	68.88	79.22
+ Reasoning-CV- <i>LLM_{SI_1}</i>	8B	76.96	87.82	83.09	86.36	80.97	83.18	68.46	84.84	69.44	80.12
+ Reasoning-CV- <i>LLM_{SI_2}</i>	8B	77.10	88.70	82.99	86.44	80.16	83.01	66.29	84.81	67.35	79.65

Table 3: Results of Reasoning-CV fine-tuned LLMs in out-of-domain test sets with gold evidence considering both *w NEI* and *w/o NEI* settings. We **bold** the best-performing LLM on each test set and underline the best-performing baseline. Results with * are collected from Tang et al. (2024a).

(i.e., the accompanying test sets for FEVEROUS, HOVER, and Healthver shown in Table 1) with gold evidence. For Healthver, we evaluate Reasoning-CV with both *w NEI* and *w/o NEI* settings. As shown in Table 2, when fine-tuning an 8B LLaMA base LLM, Reasoning-CV can significantly improve the claim verification performance stage-by-stage on all test sets with both *w NEI* and *w/o NEI* settings. Moreover, *Meta-LLaMA-3.1-8B-Instruct + Reasoning-CV-LLM_{SI_2}* can demonstrate superior average performances compared to advanced LLMs and *Decompose-Then-Verify* baselines, including PACAR, Decompose + Minicheck-7B, GPT-4o + CoT, even o1-preview. Results also indicate that the adopted *CoT-Verify* paradigm can be better compared to the *Decompose-Then-Verify* paradigm in knowledge-assisted claim verification.

4.2 Generalize to Out-of-Domain Datasets

The generalization ability of claim verification methods is significant, so we also evaluate Reasoning-CV fine-tuned LLMs on claims and knowledge collected from domains never seen in their fine-tuning. We consider five out-of-domain benchmarks and datasets, i.e., the LLM-AggrFact benchmark, the training set and the development set of SciFact, the development set and the test set of VitaminC. As shown in Table 3, utilizing an 8B base LLM, Reasoning-CV variants can achieve better results compared to Minicheck-7B and GPT-4o on all five test sets, considering both *w NEI* and *w/o NEI* settings, demonstrating its generalization ability to a wide range of application scenarios.

5 Discussion

In this section, we conduct ablation studies for the settings of the proposed Reasoning-CV. We also discuss the open book setting and insights from our experiments.

5.1 Ablation Studies

To showcase the superiority of the Reasoning-CV settings and components, we conduct a series of ablation studies on the settings of each stage.

Ablation on the First-Stage Prompt P_{V^*} The key design in the first stage of Reasoning-CV is to provide *GPT-4o* with the ground-truth veracity V^* , which may guide *GPT-4o* to effectively understand the claim and generate better reasoning paths. To determine its significance, we design an ablation variant (*w/o* Ground-truth V^* in Table below) in which we remove the prompt P_{V^*} in Equation (2), sample *GPT-4o* for up to 5 times with only P_{CE} (a similar idea is used in Wang et al. (2025)), and include the first sample with the correct veracity. As shown in Table 4, removing P_{V^*} results in a significant decrease in model performance, especially in relatively hard datasets (e.g., HOVER and Healthver), so the original setting is a better choice.

Method	FEVEROUS	HOVER	Healthver
Original	93.69	78.50	74.11
<i>w/o</i> Ground-truth V^*	92.91	77.76	70.78

Table 4: Ablation on providing the ground-truth veracity in the first stage of Reasoning-CV. We consider only the *w/o NEI* setting and take the average performance with 2 to 4 hops for HOVER.

Ablation on the Second-Stage Conditioning In the second stage of Reasoning-CV, we guide the

LLMs LLM_{SI_i} to generate with different veracity V , which can amplify the key difference between the chosen and rejected reasoning paths. To determine the effectiveness of generating conditioned on different veracities, we design an ablation variant (*w/o Conditioning* in Table below) in which we remove the prompt $P_{support}$ and P_{refute} in Equation (3) and sample LLM_{SI_i} for two times with only P_{CE} . As shown in Table 5, removing the conditioning causes worse results.

Method	FEVEROUS	HOVER	Healthver
Original	95.50	84.31	78.41
<i>w/o Conditioning</i>	94.44	80.04	75.89

Table 5: Ablation on conditional generation with different veracities in the second stage. We consider only the *w/o NEI* setting and take the average performance with 2 to 4 hops for HOVER.

Ablation on Pre-Trained LLMs We also apply Reasoning-CV on the *LlaMA-3.2-3B-Instruct* LLM to verify its applicability on pre-trained LLMs of different sizes. The results of Reasoning-CV fine-tuned LLMs are shown in Table 6, where Reasoning-CV can significantly improve the performance in both in-domain (i.e., FEVEROUS) and out-of-domain (i.e., LLM-AggrFact) test sets stage-by-stage. The fine-tuned 3B LlaMA model can achieve competitive performance compared to *GPT-4o* and Minicheck-7B on both test sets.

Method	FEVEROUS	LLM-AggrFact
GPT-4o	91.64	79.40
Minicheck-7B	89.54	76.58
LlaMA-3B	86.69	73.62
+ Reasoning-CV- LLM_{SFT}	92.59	74.35
+ Reasoning-CV- LLM_{SI_1}	94.65	75.98
+ Reasoning-CV- LLM_{SI_2}	94.75	76.20

Table 6: Ablation on using 3B pre-trained LLM *LlaMA-3.2-3B-Instruct* for Reasoning-CV.

5.2 Performance on Open Book Settings

As mentioned in Section 2.1, besides the gold evidence setting used in Table 2 and Table 3, there is an open book setting in claim verification, which requires verifiers to retrieve knowledge E from sources (e.g., the Internet). Usually, the collected evidence may not be enough to determine the veracity of claims (Vladika and Matthes, 2024), so the veracity label for gold evidence may no longer be correct and cannot be used to reliably judge which claim verification method is better. Nonetheless, we evaluate the generalization ability of Reasoning-

CV fine-tuned LLMs to the knowledge E from this setting and provide the results in Appendix E.1, where Reasoning-CV fine-tuned 8B LLMs can also outperform powerful baselines, including *GPT-4o*.

5.3 Comparison of the Second Stage to RL-based Fine-Tuning

The idea of the proposed self-improvement DPO in Reasoning-CV is quite similar to reinforcement learning-based fine-tuning (Shao et al., 2024; Liu et al., 2025b). However, the inputted claim and knowledge for fine-tuning tends to be quite long (with usually thousands of tokens). Our efforts in fine-tuning LLMs for claim verification with RL ran into Out-of-Memory problems, failing to run on a single H100-96 GPU (same device for the current Reasoning-CV). Furthermore, Reasoning-CV is able to condition the generated reasoning sequences for DPO on the different target veracities, something not done in the usual RL algorithms.

5.4 Relation to Fine-Tuning Methods for Math Reasoning

Currently, a large number of task-specific reasoning models have achieved outstanding results through distilling reasoning paths and fine-tuning LlaMA LLMs (Tu et al., 2025; Li et al., 2025; Liu et al., 2025a). The main novelty of the proposed Reasoning-CV lies in its second stage. In the second stage, the conditional LLM generation for preferences improves the quality of the reasoning paths and calibrates the LLM’s thresholds of judgments.

We believe the effectiveness of this design comes from the properties of the knowledge-assisted claim verification task, which has only two or three options for its judgment. Instead, for example, mathematical tasks tend to be open-ended with infinite possible answers, so we cannot implement the stage 2 of Reasoning-CV for these tasks. We believe that the stage 2 design in Reasoning-CV can be extended to any task with an enumerable number of answers, and we will consider it as future work.

6 Conclusion

In this paper, we propose to solve the knowledge-assisted claim verification task with the *CoT-Verify* paradigm, building the claim verification process in a long CoT reasoning path. We present Reasoning-CV to fine-tune LLMs for high-quality reasoning paths. Reasoning-CV designs to generate reliable *GPT-4o* reasoning paths in the first stage and representative preference datasets in the second stage,

with conditioning on the label. It can significantly improve the 8B LLM’s reasoning ability and get significantly better results on in-domain and out-of-domain test sets and benchmarks.

7 Limitation

As the main limitation of this paper, the reasoning ability in LLMs fine-tuned by Reasoning-CV cannot provide any help to the knowledge retrieval under the open book setting. In the future, we will try to integrate the retrieval process into the reasoning process. Moreover, as mentioned in Section 5.4, we will also try to extend Reasoning-CV to a wider range of LLM-based tasks with enumerable answer options.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. [arXiv preprint arXiv:2303.08774](#).
- Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information. [arXiv preprint arXiv:2106.05707](#).
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. [arXiv preprint arXiv:2310.11511](#).
- Hung-Ting Chen, Fangyuan Xu, Shane Arora, and Eunsol Choi. 2023. Understanding retrieval augmentation for long-form question answering. [arXiv preprint arXiv:2310.12150](#).
- Islam Eldifrawi, Shengrui Wang, and Amine Trabelsi. 2024. Automated justification production for claim veracity in fact checking: A survey on architectures and approaches. [arXiv preprint arXiv:2407.12853](#).
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. [arXiv preprint arXiv:2312.10997](#).
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. [ICLR](#), 1(2):3.
- Qisheng Hu, Quanyu Long, and Wenyang Wang. 2024. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance? [arXiv preprint arXiv:2411.02400](#).
- Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenyang Wang. 2024. Training language models to generate text with citations via fine-grained rewards. [arXiv preprint arXiv:2402.04315](#).
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. [ACM Transactions on Information Systems](#).
- Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. [arXiv preprint arXiv:2402.00559](#).
- Nazanin Jafari and James Allan. 2024. Robust claim verification through fact detection. [arXiv preprint arXiv:2407.18367](#).
- Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. Hover: A dataset for many-hop fact extraction and claim verification. [arXiv preprint arXiv:2011.03088](#).
- Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. [arXiv preprint arXiv:2303.01432](#).
- Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025. From system 1 to system 2: A survey of reasoning large language models. [arXiv preprint arXiv:2502.17419](#).
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. [arXiv preprint arXiv:2412.19437](#).
- Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. [arXiv preprint arXiv:2304.09848](#).
- Yue Liu, Hongcheng Gao, Shengfang Zhai, Jun Xia, Tianyi Wu, Zhiwei Xue, Yulin Chen, Kenji Kawaguchi, Jiaheng Zhang, and Bryan Hooi. 2025a. Guardreasoner: Towards reasoning-based llm safeguards. [arXiv preprint arXiv:2501.18492](#).
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. [arXiv preprint arXiv:2503.20783](#).
- Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2023. Expertqa: Expert-curated questions and attributed answers. [arXiv preprint arXiv:2309.07852](#).

- Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. [arXiv preprint arXiv:2305.14251](#).
- Yassine Mrabet Mourad Sarouti, Asma Ben Abacha and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In [EMNLP](#).
- Fan Nie, Xiaotian Hou, Shuhang Lin, James Zou, Huaxiu Yao, and Linjun Zhang. 2024. Facttest: Factuality testing in large language models with statistical guarantees. [arXiv preprint arXiv:2411.02603](#).
- Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. [arXiv preprint arXiv:2305.12744](#).
- Dorian Quelle and Alexandre Bovet. 2024. The perils and promises of fact-checking with large language models. [Frontiers in Artificial Intelligence](#), 7:1341697.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. [Advances in Neural Information Processing Systems](#), 36:53728–53741.
- Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. [arXiv preprint arXiv:2103.08541](#).
- Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. [arXiv preprint arXiv:2305.15294](#).
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. [arXiv preprint arXiv:2402.03300](#).
- Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. [arXiv preprint arXiv:2409.12183](#).
- Marcella Tambuscio, Giancarlo Ruffo, Alessandro Flammini, and Filippo Menczer. 2015. Fact-checking effect on viral hoaxes: A model of misinformation spread in social networks. In [Proceedings of the 24th international conference on World Wide Web](#), pages 977–982.
- Liyan Tang, Tanya Goyal, Alexander R Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin F Rousseau, and Greg Durrett. 2022. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. [arXiv preprint arXiv:2205.12854](#).
- Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. Minicheck: Efficient fact-checking of llms on grounding documents. [arXiv preprint arXiv:2404.10774](#).
- Liyan Tang, Igor Shalyminov, Amy Wing-meい Wong, Jon Burnsky, Jake W Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, and 1 others. 2024b. Tofueval: Evaluating hallucinations of llms on topic-focused dialogue summarization. [arXiv preprint arXiv:2402.13249](#).
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. [arXiv preprint arXiv:2302.13971](#).
- Songjun Tu, Jiahao Lin, Xiangyu Tian, Qichao Zhang, Linjing Li, Yuqian Fu, Nan Xu, Wei He, Xiangyuan Lan, Dongmei Jiang, and 1 others. 2025. Enhancing llm reasoning with iterative dpo: A comprehensive empirical investigation. [arXiv preprint arXiv:2503.12854](#).
- Juraj Vladika and Florian Matthes. 2024. Comparing knowledge sources for open-domain scientific claim verification. [arXiv preprint arXiv:2402.02844](#).
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. [arXiv preprint arXiv:2004.14974](#).
- Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. [arXiv preprint arXiv:2310.05253](#).
- Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, and 1 others. 2023. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. [arXiv preprint arXiv:2311.09000](#).
- Zhengren Wang, Jiayang Yu, Dongsheng Ma, Zhe Chen, Yu Wang, Zhiyu Li, Feiyu Xiong, Yanfeng Wang, Linpeng Tang, Wentao Zhang, and 1 others. 2025. Rare: Retrieval-augmented reasoning modeling. [arXiv preprint arXiv:2503.23513](#).
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. [Advances in neural information processing systems](#), 35:24824–24837.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, and 1 others. 2024. Long-form factuality in large language models. [arXiv preprint arXiv:2403.18802](#).

Xia Zeng and Arkaitz Zubiaga. 2024. Maple: Micro analysis of pairwise language evolution for few-shot claim verification. [arXiv preprint arXiv:2401.16282](#).

Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. [arXiv preprint arXiv:2310.00305](#).

Xuan Zhang and Wei Gao. 2024. Reinforcement retrieval leveraging fine-grained feedback for fact checking news claims with black-box llm. [arXiv preprint arXiv:2404.17283](#).

Xiaoyan Zhao, Lingzhi Wang, Zhanghao Wang, Hong Cheng, Rui Zhang, and Kam-Fai Wong. 2024. Pacar: Automated fact-checking with planning and customized action reasoning using large language models. In [Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation \(LREC-COLING 2024\)](#), pages 12564–12573.

Yingjie Zhu, Jiasheng Si, Yibo Zhao, Haiyang Zhu, Deyu Zhou, and Yulan He. 2023. Explain, edit, generate: rationale-sensitive counterfactual data augmentation for multi-hop fact verification. [arXiv preprint arXiv:2310.14508](#).

A Prompts for Claim Verification

In this section, we display all the prompts adopted for Reasoning-CV fine-tuning and the fine-tuned LLM. We use the Veracity Generation Prompt P_{CE} in generating the veracity V for claim C and knowledge E . The Conditioning Prompt P_V is used to generate *GPT-4o* labels for SFT (the first stage) by providing the ground-truth V^* , it is also adopted to lead LLMs for preference pairs in the self-improvement DPO stage.

We list our prompt and introduce them in details as follows:

The Veracity Generation Prompt P_{CE} for the w/o NEI Setting

System:

Task: Validate the following claim using the provided context.

Your goal is to determine whether the claim can be supported by the context. Choose between "support" or "refute".

Instructions: 1. Analyze the claim step by step, verifying each crucial component in the claim as they appear.

2. Structure your reasoning on crucial components in the claim in detailed steps, from 1 to a maximum of 10. Make sure each step is the smallest possible logical unit necessary for validation.

3. Ensure that your reasoning correlates consistently with your conclusion. Use "##" to format each step clearly, e.g., "## Reasoning Step 1".

4. Finally, conclude with either "support" or "refute" enclosed in a pair of curly braces, noting the overall judgment regarding the claim.

User:

Context: Life Goes On is a song recorded by American singer Fergie for her second studio album, Double Dutchess (2017). It was released as single on November 11, 2016, by Interscope and will.i.am Music Group. The song serves as the third single from Fergie's second studio album, following M.I.L.F. \$ Life Goes On was written by Fergie, Tristan Prettyman, Keith Harris and Toby Gad. M.I.L.F. \$(pronounced MILF money) is

a song recorded by American singer Fergie for her second studio album, Double Dutchess (2017). It was produced by Polow da Don and released as the second single from the record following L.A. Love (La La) on July 1, 2016 by Interscope and will.i.am Music Group. It debuted at number 34 on the US Billboard Hot 100 with 65,000 in first-week sales.

Claim: The song recorded by Fergie that was produced by Polow da Don and was followed by Life Goes On was M.I.L.F\$.

We select a claim in the HOVER dataset to provide the example above for the *w/o NEI* setting. The above-mentioned prompt is used in:

- In training, providing claims and knowledge with CoT instructions (is used together with P_V).
- In testing, generating *Cot-Verify* results for data under the *w/o NEI* setting.

The Veracity Generation Prompt P_{CE} for the *w NEI* Setting

System:

Task: Validate the following claim using the provided context.

Your goal is to determine whether the claim can be supported with the context. Choose between "support", "refute", or "not enough information".

Instructions:

1. Analyze the claim step by step, verifying each crucial component in the claim as they appear.
2. Structure your reasoning on crucial components in the claim in detailed steps, from 1 to a maximum of 10. Make sure each step is the smallest possible logical unit necessary for validation.
3. Ensure that your reasoning correlates consistently with your conclusion. Use "##" to format each step clearly, e.g., "## Reasoning Step 1".
4. Finally, conclude with "support", "refute", or "not enough information" enclosed in a pair of curly braces, noting the overall judgment regarding the claim.

ment regarding the claim.

User:

Context: In this study, we collected blood from COVID-19 patients who have recently become virus-free and therefore were discharged, and analyzed their SARS-CoV-2-specific antibody and T cell responses. We observed SARS-CoV-2-specific humoral and cellular immunity in the patients. Both were detected in newly discharged patients, suggesting both participate in immune-mediated protection to viral infection.

Claim: For most patients, COVID-19 begins and ends in their lungs, because like the flu, coronaviruses are respiratory diseases.

We select a claim in the Healthver dataset to provide the example above for the *w NEI* setting. The above-mentioned prompt is used in:

- In training, providing claims and knowledge with CoT instructions (is used together with P_V).
- In testing, generating *Cot-Verify* results for data under the *w NEI* setting.

With the Ground-truth Veracity Prompt P_V (the *w/o NEI* Setting Case as an Example)

System:

Task: Validate the following claim using the provided context.

Your goal is to determine whether the claim can be supported by the context. Choose between "support" or "refute".

Instructions: 1. Analyze the claim step by step, verifying each crucial component in the claim as they appear.

2. Structure your reasoning on crucial components in the claim in detailed steps, from 1 to a maximum of 10. Make sure each step is the smallest possible logical unit necessary for validation.

3. Ensure that your reasoning correlates consistently with your conclusion. Use "##" to format each step clearly, e.g., "## Reasoning Step 1".

4. Finally, conclude with either "support" or "refute" enclosed in a pair of curly braces, noting the overall judgment regarding the claim.

The ground truth is

—
Answer: support. You must generate results that match ground truth.

User:

Context: Life Goes On is a song recorded by American singer Fergie for her second studio album, Double Dutchess (2017). It was released as single on November 11, 2016, by Interscope and will.i.am Music Group. The song serves as the third single from Fergie's second studio album, following M.I.L.F. \$. Life Goes On was written by Fergie, Tristan Prettyman, Keith Harris and Toby Gad. M.I.L.F. \$(pronounced MILF money) is a song recorded by American singer Fergie for her second studio album, Double Dutchess (2017). It was produced by Polow da Don and released as the second single from the record following L.A. Love (La La) on July 1, 2016 by Interscope and will.i.am Music Group. It debuted at number 34 on the US Billboard Hot 100 with 65,000 in first-week sales.

Claim: The song recorded by Fergie that was produced by Polow da Don and was followed by Life Goes On was M.I.L.F.\$.

We select a claim in the HOVER dataset to provide the example above for P_V . This prompt is used in the two stages of training, providing the ground-truth V^* (noted as P_{V^*}) in the first stage and guiding the conditional generation for each veracity option (noted as $P_{support}$, P_{NEI} , and P_{refute}) in the second stage.

B Training Dataset Examples

To help the comprehension of the Reasoning-CV fine-tuning process, in this section, we provide examples for the training data in two stages as follows:

Example of SFT Training data in stage

1. (Output of GPT-4o with CoT prompt (P_{CE}) and conditioning prompt (P_{V^*}))

"instruction": Task: Validate the following claim using the provided context.

Your goal is to determine whether the claim can be supported by the context. Choose between "support" or "refute".

Instructions: 1. Analyze the claim step by step, verifying each crucial component in the claim as they appear.

2. Structure your reasoning on crucial components in the claim in detailed steps, from 1 to a maximum of 10. Make sure each step is the smallest possible logical unit necessary for validation.

3. Ensure that your reasoning correlates consistently with your conclusion. Use "##" to format each step clearly, e.g., "## Reasoning Step 1".

4. Finally, conclude with either "support" or "refute" enclosed in a pair of curly braces, noting the overall judgment regarding the claim.

"input": Context: Joseph Nassise (born 1968) is a New York Times and USA Today bestselling American urban fantasy writer and the author of more than forty novels.

Joseph Nassise lives with his wife and four children in Arizona. Joseph Nassise is the author of the internationally bestselling Templar Chronicles series, the Jeremiah Hunt Chronicle, the Great Undead War series, as well as several books for Gold Eagle's Rogue Angel line.

Joseph Nassise served as the president of the Horror Writers Association from 2002 to 2005 and a Trustee of the same from 2008 to 2010.

Claim: Joseph Nassise is a New York Times and USA Today bestselling American urban fantasy writer and the author residing in Arizona who wrote Templar Chronicles series, the Jeremiah Hunt Chronicle and the Great Undead War series and served as the president of the Horror Writers Association from 2002 to 2005.

"output": "## Reasoning Step 1: Verify if Joseph Nassise is a New York Times and USA Today bestselling American urban fantasy writer.

- The context states that Joseph Nassise is a New York Times and USA Today bestselling American urban fantasy writer.

Reasoning Step 2: Verify if Joseph Nassise resides in Arizona.

- The context confirms that Joseph Nassise lives with his wife and four children in Arizona.

Reasoning Step 3: Verify if Joseph Nassise wrote the Templar Chronicles series.

- The context mentions that Joseph Nassise is the author of the internationally bestselling Templar Chronicles series.

Reasoning Step 4: Verify if Joseph Nassise wrote the Jeremiah Hunt Chronicle.

- The context states that Joseph Nassise is the author of the Jeremiah Hunt Chronicle.

Reasoning Step 5: Verify if Joseph Nassise wrote the Great Undead War series.

- The context confirms that Joseph Nassise is the author of the Great Undead War series.

Reasoning Step 6: Verify if Joseph Nassise served as the president of the Horror Writers Association from 2002 to 2005.

- The context indicates that Nassise served as the president of the Horror Writers Association from 2002 to 2005.

Reasoning Step 7: Verify if there are any discrepancies or missing components in the claim.

- The claim accurately reflects the information provided in the context without any discrepancies or missing components.

Based on the above steps, the claim is fully supported by the context.

{support}"

Instructions: 1. Analyze the claim step by step, verifying each crucial component in the claim as they appear.

2. Structure your reasoning on crucial components in the claim in detailed steps, from 1 to a maximum of 10. Make sure each step is the smallest possible logical unit necessary for validation.

3. Ensure that your reasoning correlates consistently with your conclusion. Use "##" to format each step clearly, e.g., "## Reasoning Step 1".

4. Finally, conclude with either "support" or "refute" enclosed in a pair of curly braces, noting the overall judgment regarding the claim.

"input": Context: The Boxer is a medium-sized, short-haired breed of dog, developed in Germany. The coat is smooth and tight-fitting; colours are fawn or brindled, with or without white markings, and white. Boxers are brachycephalic (they have broad, short skulls), have a square muzzle, mandibular prognathism (an underbite), very strong jaws, and a powerful bite ideal for hanging on to large prey. The Boxer was bred from the Old English Bulldog and the now extinct Bullenbeisser which became extinct by crossbreeding rather than by a decadence of the breed. The purpose of the crossbreeding was the wish to eliminate the excessive white color of the breed, and the necessity of producing thousands of dogs for one of the most popular breeds in the world. The Boxer is part of the Molosser group. This group is a category of solidly built, large dog breeds that all descend from the same common ancestor, the large shepherd dog known as a Molossus. The Boxer is a member of the Working Group.

Can de Palleiro or Pastor Galego also known as Galician Shepherd Dog or Galician Palleiro is a dog breed originating in Galicia (Spain).

The Bullenbeisser (also known as the German Bulldog) was a breed of dog known for its strength and agility. The breed was closely related to the Bärenbeisser

Example of DPO pairs in stage 2.

"instruction": Task: Validate the following claim using the provided context.

Your goal is to determine whether the claim can be supported by the context. Choose between "support" or "refute".

(some believe that the two breeds were the same; the names mean bull-biteränd bear-biter, respectively), and the Boxer. It was, in all its aspects, similar to the present Alano Español (Spanish Bulldog) and very alike to the Dogo Argentino, not only in aspect, but also in usage. There were two regional varieties, the Brabanter Bullenbeisser and the Danziger Bullenbeisser. The breed is now extinct.

Claim: The Can de Palleiro is a dog that originated in Wong, not the dog that the Old English Bulldog was bred with to make the Boxer (dog).

"chosen": "## Reasoning Step 1: The claim states that the Can de Palleiro is a dog that originated in Wong.

- The context specifies that the Can de Palleiro, also known as the Galician Shepherd Dog or Galician Palleiro, is a dog breed originating in Galicia, Spain, not Wong.
- ## Reasoning Step 2: The claim states that the Can de Palleiro is not the dog that the Old English Bulldog was bred with to make the Boxer.
- The context indicates that the Boxer was bred from the Old English Bulldog and the now extinct Bullenbeisser, not the Can de Palleiro.

Reasoning Step 3: Verify the relationship between the Can de Palleiro and the Boxer.

- The context does not mention any relationship between the Can de Palleiro and the Boxer, further supporting that the Can de Palleiro was not involved in the breeding of the Boxer.

Reasoning Step 4: Confirm the accuracy of the claim based on the context.

- The context supports the claim that the Can de Palleiro is not the dog that the Old English Bulldog was bred with to make the Boxer, as it was the Bullenbeisser that was used.

Reasoning Step 5: Evaluate the overall claim.

- The claim is partially incorrect regarding the origin of the Can de Palleiro, but correct in stating that it was not involved in the breeding of the Boxer.

Based on the analysis, the claim is partially correct but contains an incorrect statement about the origin of the Can de Palleiro. Therefore, the claim is overall refuted.

{refute}" ,

"rejected": "## Reasoning Step 1: The claim states that the Can de Palleiro is a dog that originated in Wong. The context does not mention Wong as the origin of the Can de Palleiro. Instead, it states that the Can de Palleiro, also known as the Galician Shepherd Dog, originated in Galicia, Spain.

Reasoning Step 2: The claim states that the Can de Palleiro is not the dog that the Old English Bulldog was bred with to make the Boxer. The context specifies that the Boxer was bred from the Old English Bulldog and the now extinct Bullenbeisser, not the Can de Palleiro.

Reasoning Step 3: The context confirms that the Bullenbeisser was one of the breeds used to create the Boxer, and it was not the Can de Palleiro. Therefore, the claim that the Can de Palleiro was not used in the breeding of the Boxer is consistent with the context.

Reasoning Step 4: The context does not provide any information linking the Can de Palleiro to the breeding of the Boxer, further supporting the claim that it was not involved in the creation of the Boxer breed.

Reasoning Step 5: The claim is consistent with the context in stating that the Can de Palleiro is not related to the breeding of the Boxer, as the context clearly attributes the breeding of the Boxer to the Old English Bulldog and the Bullenbeisser.

{support}"

C Case Study

As the complement to Figure 4, in this section, we show the cases of the LLM's drawbacks after the first stage of SFT fine-tuning.

Case 1: Fault in judging the acceptance threshold.

Claim: Many of the works of Dallas John Baker are gay-themed, though not all of them, and some of his short stories are set in Australia.

Context: Dallas John Baker (born Dallas John Anguish Baker on 19 February 1968) is an Australian writer and academic noted for poetry, short stories and travel writing whose work deals with themes of alienation, otherness and sexuality. **Some** have gay content, but not all of them. Local readers will be interested in the final section of stories, set in Australia with locales ranging from Byron Bay to the Darling Downs.

Ground-truth Veracity: Support

Context: ## Reasoning Step 1: The claim states that many of the works of Dallas John Baker are gay-themed. The context mentions that some of his works have gay content, but it does not specify that many of his works are gay-themed. Therefore, this part of the claim is not fully supported by the context.

...

{refute}

In this case, the LLM model determines a claim that can be validated by context knowledge as ‘*refute*’. Based on the output content, we can find that the wrong output is because the LLM believes that the paraphrase from “some” to “many” can be a fundamental fault, which does not match the human judging threshold. Therefore, we design Stage 2 of Reasoning-CV to align the LLM judgment with a reasonable threshold.

Case 2: Inconsistent reasoning paths.

Claim: The suburb of Louisville is the headquarters of the business founded by John H Schnatter. It is a rule-class city of 26,595 and it contains the neighborhood Plainview.

Context: Plainview is a neighborhood in the city of Jeffersontown, Kentucky and is located on the former site of the Plainview Dairy Farm. It is a Planned Unit Development which includes housing, workplaces, and shopping within a single development. John H. Schnatter (born November 22, 1961) is an American entrepreneur and the founder, CEO, and spokesman of Papa John's International, Inc.

Jeffersontown is a home rule-class city in

Jefferson County, Kentucky, United States. The population was 26,595 at the 2010 U.S. census.

Papa John's Pizza is an American restaurant franchise company. It runs the third largest take-out and pizza delivery restaurant chain in the United States, with headquarters in Jeffersontown, Kentucky, a suburb of Louisville.

Ground-truth Veracity: Support

Context:

Reasoning Step 2: From the Claim, Jeffersontown is a suburb of Louisville, Kentucky.
...

Reasoning Step 5: The claim correctly states that the headquarters of the business founded by John H. Schnatter is in Jeffersontown, which contains the neighborhood Plainview. However, it incorrectly identifies Louisville as the suburb.

{refute}

In this case, the LLM model also determines a claim that can be validated by context knowledge as ‘*refute*’. Based on the output content, we can find that the wrong output is because the LLM performs a hallucination and suddenly changes its previous correct assertion “Jeffersontown is a suburb of Louisville” to “it incorrectly identifies Louisville as the suburb”. Therefore, we design stage 2 of Reasoning-CV to improve the consistency of the *CoT-Verify* generation.

D Details of Reasoning-CV

In this section, we provide necessary details of the proposed Reasoning-CV, including the fine-tuning procedure for the w dataset Healthver.

D.1 Fine-Tuning Details for Claims in Healthver

In the training dataset of Reasoning-CV, FEVEROUS and HOVER support only the w/o NEI setting (i.e., have ‘support’ and ‘refute’ options in their labels), and Healthver supports the w NEI setting (i.e., has ‘support’, ‘NEI’, and ‘refute’ options in its labels).

So, in the second stage of Reasoning-CV, we enumerate two options for FEVEROUS and HOVER (as shown in the main text), and enumerate three options for Healthver. The detailed process of building $\mathcal{D}_{SI_{i-1}}$ for Healthver is as follows:

$$\begin{aligned} R_{support} &= LLM_{SI_{i-1}}(P_{CE}, P_{support}), \\ R_{NEI} &= LLM_{SI_{i-1}}(P_{CE}, P_{NEI}), \\ R_{refute} &= LLM_{SI_{i-1}}(P_{CE}, P_{refute}). \end{aligned} \quad (5)$$

$$\begin{aligned} \mathcal{D}_{SI_{i-1}} = \Big\{ & [V^* \in R_{support}] (R_{refute} \prec R_{support}), \\ & [V^* \in R_{support}] (R_{NEI} \prec R_{support}), \\ & [V^* \in R_{NEI}] (R_{refute} \prec R_{NEI}), \\ & [V^* \in R_{NEI}] (R_{support} \prec R_{NEI}), \\ & [V^* \in R_{refute}] (R_{support} \prec R_{refute}), \\ & [V^* \in R_{support}] (R_{NEI} \prec R_{refute}) \Big\}, \end{aligned} \quad (6)$$

E More Experiments

E.1 Reasoning-CV Performance In the Open Book Setting

In the Open Book setting of adopted datasets, the claim verification method is required to retrieve knowledge E from the Internet (Wang and Shu, 2023). To achieve this, as shown in Figure 5, we follow the idea from Zhao et al. (2024), break the claim into sub-claims, and then retrieve the most related paragraph from Google with the Serper API, taking each sub-claim as the query. We use the prompts from Min et al. (2023) for the claim decomposition and decontextualization. To build an 8B open book claim verification system, we fine-tune another *Meta-LlaMA-3-8B-Instruct* LLM by SFT based on the *GPT-4o* output of claim decomposition and decontextualization for independent sub-claims. As shown in Table 7, where we use the

same collected knowledge for all baselines except Zhao et al. (2024). When using the knowledge E obtained from the open book setting, Reasoning-CV still outperforms *GPT-4o* in the average performance, demonstrating the generalization ability of the obtained model to various evidence settings.

E.2 Comparison to Claim Verification Methods with Adaptive Retrieval

As mentioned in related work, ProgramFC (Pan et al., 2023) is a typical claim verification method with adaptive retrieval. In this part, we compare the proposed Reasoning-CV fine-tuned LLMs with ProgramFC, as shown in Table 8, Reasoning-CV leads in 7 out of 8 test sets, demonstrating the power of the proposed Reasoning-CV and its adopted *CoT-Verify* paradigm.

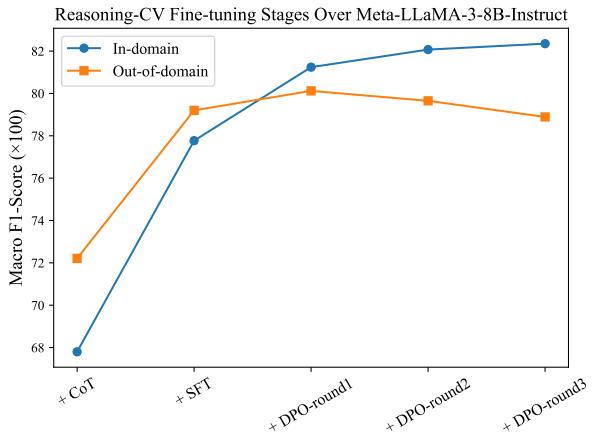


Figure 6: The performance curves for in-domain and out-of-domain datasets as the Reasoning CV fine-tuning progresses.

E.3 Performance over the Number of Rounds for Self-improvement DPO

Table 2 and Table 3 preliminarily demonstrate the performance changes of Reasoning-CV fine-tuned LLMs as the number of self-improvement DPO rounds increases. In this section, we use the average performance over dataset and settings (i.e., “Avg.” in Table 2 and Table 3) to evaluate the in-domain and out-of-domain performances, respectively, and draw a performance curve over Reasoning-CV stages and DPO rounds in Figure 6. As the number of DPO rounds increases from 0 (i.e., + SFT) to 3, Reasoning CV fine-tuned LLMs will perform significantly better in the in-domain. However, more than one round of Self-improvement DPO may impair LLM’s effectiveness on out-of-domain claims, and *Meta-LLaMA-3-*

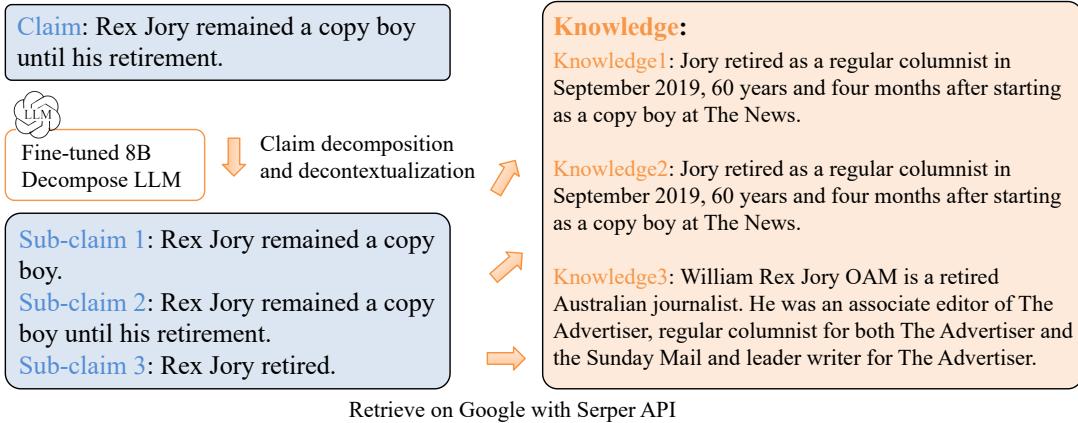


Figure 5: The evidence retrieval process in the open book setting of Reasoning-CV. We fine-tune another 8B LLaMA model for the claim decomposition and decontextualization tasks in Min et al. (2023). We retrieve the top paragraph from Google by the Serper API, taking each sub-claim as a query.

Open Book							
Dataset	Param Size	FEVEROUS	HOVER (2-hop)	HOVER (3-hop)	HOVER (4-hop)	Healthver	
Method	Settings	w/o NEI	w/o NEI	w/o NEI	w/o NEI	w/o NEI	w NEI
GPT-4o	N/A	<u>74.27</u>	68.37	57.06	52.42	<u>55.30</u>	<u>43.27</u>
Minicheck-7B	7B	74.03	65.90	46.75	40.90	52.47	-
Established Decompose-Then-Verify Methods							
PACAR*	N/A	72.61	73.13	64.07	63.82	-	-
FactScore	N/A	69.29	61.10	46.11	43.50	53.86	41.40
Decompose + Minicheck-7B	7B	69.87	61.14	41.71	36.06	51.98	-
Reasoning-CV for High-Quality CoT-Verify							
Meta-LLaMA-3-8B-Instruct	8B	78.80	65.75	60.06	58.44	48.93	32.36
+ CoT Prompt	8B	76.53	63.59	53.98	54.92	54.38	36.82
+ Reasoning-CV-LLM _{SFT}	8B	79.86	72.29	59.16	51.24	55.59	41.88
+ Reasoning-CV-LLM _{SI₁}	8B	80.32	72.47	59.64	49.20	54.20	43.35
+ Reasoning-CV-LLM _{SI₂}	8B	80.22	71.98	61.44	49.86	54.05	41.60
							59.86

Table 7: Reasoning-CV fine-tuned LLMs in in-domain test sets with open book considering both w NEI and w/o NEI settings. We **bold** the best-performing LLM on each test set and underline the best-performing baseline. Results with * are collected from Zhao et al. (2024), with possibly different knowledge for each claim.

Method	Gold Evidence				Open Book				Avg.	
	FEVEROUS	HOVER			FEVEROUS	HOVER				
		2-hop	3-hop	4-hop		2-hop	3-hop	4-hop		
GPT-4o	91.64	79.48	75.83	73.52	74.27	68.37	57.06	52.42	71.57	
ProgramFC*	91.77	74.10	66.13	65.69	67.80	69.36	60.63	59.16	69.33	
Meta-LLaMA-3-8B-Instruct	90.34	74.24	67.21	66.39	78.80	65.75	60.06	58.44	70.15	
+ Reasoning-CV-LLM _{SFT}	93.72	81.26	78.70	75.65	79.86	72.29	59.16	51.24	73.99	
+ Reasoning-CV-LLM _{SI₂}	95.50	85.97	83.93	83.02	80.22	71.98	61.44	49.86	76.49	

Table 8: Reasoning-CV fine-tuned LLMs in in-domain test sets with both gold evidence and open book. We bold the best result for each test set. We use the report results for ProgramFC.

Base LLM	Stage1		Stage2-Round1		Stage2-Round2		Total
	GPT-4o labeling	LlaMA SFT	LLM _{SFT} labeling	LLM _{SFT} DPO	LLM _{SI₁} labeling	LLM _{SI₁} DPO	
LLaMA-3.2-3B-Instruct	12h	6h	6h	3.2h	7h	3.2h	37.4h
Meta-LLaMA-3-8B-Instruct	12h	8.5h	3h	7.5h	3.2h	9h	43.2h

Table 9: Reasoning-CV fine-tuned LLMs in in-domain test sets with both gold evidence and open book. We bold the best result for each test set. We use the report results for ProgramFC.

8B-Instruct + Reasoning-CV-LLM_{SI₁} can be the best claim verification LLM for the out-of-domain.

E.4 Time Consumption for Stages

To evaluate the efficiency of Reasoning-CV finetuning, we list the time consumption for Reasoning-CV on a single H100-96B GPU. As shown in Table 9, the entire Reasoning-CV fine-tuning process can be completed in a relatively short amount of time (which will take less than two days).

F Dataset & Baseline & Licenses

This section will list all the datasets and baseline methods used in this article, introducing their features and implementation details, and providing their sources and licenses.

F.1 Dataset

This paper involves three datasets with complex claims (i.e., FEVEROUS, HOVER, and Healthver) and three datasets with relatively simple claims (i.e., LLM-AggrFact, SciFact, and VitaminC).

Healthver, SciFact, and VitaminC support both the *w* NEI setting and *w/o* NEI setting (taking the label ‘NEI’ as a sub-case of ‘refute’).

FEVEROUS: FEVEROUS (Aly et al., 2021) is a famous dataset for knowledge-assisted claim verification over unstructured and structured data, collecting evidence from sentences or cells from tables in Wikipedia. We adopt the setup from the previous methods Pan et al. (2023); Zhao et al. (2024), only considering claims that require sentence evidence.

HOVER: HOVER (Jiang et al., 2020) includes claims that can only be solved with multi-hop reasoning. It is divided into subsets based on the number of reasoning “hops” needed for claim verification (e.g., HOVER(2-hop) in Table 2 for claims 2-hop reasoning).

Healthver: Healthver (Mourad Sarrouti and Demner-Fushman, 2021) collects claims from real-world scenarios and knowledge from scientific articles. Unlike most claim verification datasets, where contradicted claims are usually just the negation of the supported ones, in Healthver, contradicted claims are themselves extracted from real-world claims, so the claims in this dataset are more challenging compared to other datasets (Jafari and Allan, 2024).

LLM-AggrFact: LLM-AggrFact dataset is a benchmark proposed in Tang et al. (2024a), which is an aggregation of 10 existing datasets

with relatively simple claims, including AggrFact (Tang et al., 2022), TofuEval (Tang et al., 2024b), ClaimVerify (Liu et al., 2023), LGQA (Chen et al., 2023), ExpertQA (Malaviya et al., 2023), Reveal (Jacovi et al., 2024), FactCheck-GPT (Wang et al., 2023), and WICE (Kamoi et al., 2023). For this dataset, decomposition of sentences into atomic facts is not necessary to achieve this high performance (Tang et al., 2024a). **Importantly**, this dataset considers the scenario of detecting LLM output hallucination. Evaluating claim verification methods on this extensive dataset can effectively demonstrate their generalization ability.

SciFact: SciFact (Wadden et al., 2020) focuses on claims and knowledge in the science domain. We use the training set and the development set for evaluation because these sets provide gold evidence.

VitaminC: Evidence sources often change over time as more information is gathered and revised. To adapt to this change, models must be sensitive to subtle differences in supporting evidence. VitaminC (Schuster et al., 2021) is proposed to evaluate the ability of claim verification methods in this situation. We consider both its development set and testing set for evaluation.

The number of claims in the test set of each dataset is listed in Table 10.

F.2 Baseline

This paper includes advanced black-box LLMs and established claim verification methods as baselines, including Minicheck (Tang et al., 2024a), **GPT-4o**, **GPT-4o+CoT**, and **o1-preview**, FactScore (Min et al., 2023), PACAR (Zhao et al., 2024), and **Decompose + Minicheck** (Tang et al., 2024a). We also include ProgramFC (Pan et al., 2023) in the Appendix.

We use the OpenAI API for **GPT-4o**, **GPT-4o+CoT**, and **o1-preview**.

We reimplement the decomposition and decontextualization process in **FactScore** with the provided prompts (Min et al., 2023) and validate the veracity of sub-claims by prompting LLMs with knowledge. Running FactScore usually requires tens of LLM calls for each complex claim, so we can only use *GPT-4o-mini* to implement it. For the *w/o* NEI setting, we judge the veracity of claims with Eq. 1. For the *w* NEI setting, with a similar spirit, we judge the claim as ‘NEI’ if and only if there are ‘NEI’ sub-claims but no ‘refute’ sub-claims.

	In-Domain				
Dataset	FEVEROUS	HOVER(2-hop)	HOVER(3-hop)	HOVER(4-hop)	Healthver
# of Eval	2962	1126	1835	1039	1823
Out-of-Domain					
Dataset	LLM-AggreFact	SciFact-train	SciFact-dev	VitaminC-dev	VitaminC-test
# of Eval	29320	809	300	63054	55197

Table 10: The number of claims in each dataset.

Resources	Type	License	URL
Minicheck	Code	Apache-2.0 license	https://github.com/Liyan06/MiniCheck
Minicheck-1B	LLM	MIT License	https://huggingface.co/lytang/MiniCheck-Flan-T5-Large
Minicheck-7B	LLM	CC BY-NC 4.0	https://huggingface.co/bespokelabs/Bespoke-MiniCheck-7B
FEVEROUS	Dataset	MIT License	https://github.com/teacherpeterpan/ProgramFC
HOVER	Dataset	MIT License	https://github.com/teacherpeterpan/ProgramFC
Healthver	Dataset	Available online	https://github.com/sarrouti/Healthver
LLM-AggreFact	Benchmark	CC BY-NC 4.0	https://huggingface.co/datasets/lytang/LLM-AggreFact
SciFact	Dataset	CC BY-NC 4.0 & ODC-By 1.0	https://github.com/allenai/scifact?tab=readme-ov-file
VitaminC	Dataset	CC BY-SA 3.0	https://huggingface.co/datasets/tals/vitaminc

Table 11: A summary of licenses.

For **Minicheck** (Tang et al., 2024a), we use their model for verification results. **Minicheck** (Tang et al., 2024a) suggests users to break complex claims first, so to break complex claims into easier ones, in Table 2, we include **Decompose + Minicheck**. **Decompose + Minicheck** utilizes *GPT-4o-mini* and prompts from FactScore (Min et al., 2023) for claim decomposition and decontextualization. We observe consistent conclusions with Hu et al. (2024) and Tang et al. (2024a) that breaking claims does not improve the effectiveness for claim verification, which reinforces the correctness of the adopted *CoT-Verify* paradigm in Reasoning-CV. It is worth noting that on the dataset with complex claims (shown in Table 2), we can observe the conclusion shown in Tang et al. (2024a), a separate decomposition step will not lead to better claim verification results.

For **PACAR** (Zhao et al., 2024), it reaches the current state-of-the-art claim verification performances with the *Decompose-Then-Verify* paradigm, but we cannot get their prompts and implementations. So, we report their results in articles (Zhao et al., 2024).

ProgramFC (Pan et al., 2023) is a typical claim verification method for adaptive retrieval and solving ideas. We also use the report results in the Appendix for ProgramFC.

F.3 License

The licenses and URL of baselines are listed in Table 11.